Introducing HCatalog

HCatalog is a table and storage management layer for
Hadoop that enables users with different data processing tools,
such as Pig and MapReduce, to more easily read and
write data on the grid.
HCatalog's table abstraction presents users with a relational
view of data in the Hadoop Distributed File System, HDFS,
and ensures that users need not worry about where or
in what format their data is stored, RCFile format,
text files, sequence files, or ORC files.
HCatalog is built on top of the Hive metastore, and
incorporate Hive's DDL.
It also provides read and write interfaces for Pig and
MapReduce, and uses Hive's command-line interface for
issuing data definition and metadata exploration commands.
HCatalog supports reading and writing files in any format for
which a SerDes, or otherwise, serializer/deserializer,
can be written.
By default, HCatalog supports RCFile, JSON, CSV, and
SequenceFile, and ORC file formats.
To use a custom format, you must provide the input format,
output format, and SerDes.
There are some benefits of HCatalog,
such as it frees the user from having to know where the data is
stored with the table abstraction.
It also enables notifications of data availability,
and provides visibility for data cleaning and archiving tools.
HCatalog provides read and write interfaces for Pig and
MapReduce.
And it uses Hive's command-line interface for
issuing data definition and metadata exploration commands.
The HCatalog interface for
Pig consists of the HCatLoader and HCatStorer,
which implement the Pig load and store interfaces, respectively.
HCatLoader accepts a table to read data from.
You can indicate which partitions to scan by
immediately following the load statement with a partition
filter statement.
HCatStorer accepts a table to write to, and optionally,
a specification of partition keys to create a new partition.
You can write to a single partition by specifying
the partition keys and values in the store clause.
And you can write to multiple partitions if the partition
keys are columns in the data being stored.
The HCatalog interface for MapReduce,

which is the HCatInputFormat and HCatOutputFormat,
is an implementation of Hadoop InputFormat and OutputFormat.
HCatInputFormat accepts a table to read data from, and
optionally, a selection predicate to indicate which
partitions to scan.
HCatOutputFormat accepts a table to write to, and optionally,
a specification of partition keys to create a new partition.
The following factors will help you decide whether to
incorporate HCatalog in your HDInsight solution.
It makes it easy to abstract the data storage location, format,
and schema from the code used to process it.
HCatalog minimizes fragile dependencies between scripts and
complex data processing solutions where the same data is
processed by multiple tasks.
It also enables notification of data availability, making it
easier to write applications that perform multiple jobs.
It is easy to incorporate into solutions that includes Hive and
Pig scripts, requiring very little extra code.
However, if you use only Hive scripts and
queries, or you are creating a one-shot solution for
experimentation purposes and do not intend to use it again,
HCatalog is unlikely to provide any benefit.
Files in JSON, SequenceFile, CSV,
and RC format can be read and written to by default.
And a custom serializer/deserializer
component can be used to read and
write files in other formats.
Additional effort is required to use HCatalog in custom
map/reduce components because you must create your own custom
load and store functions.